Anomaly Guided Policy Learning from Imperfect Demonstrations*

Zi-Xuan Chen, Xin-Qiang Cai, Yuan Jiang*, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology, Nanjing University, China
chenzx@lamda.nju.edu.cn,cai@ms.k.u-tokyo.ac.jp
jiangy@lamda.nju.edu.cn,zhouzh@lamda.nju.edu.cn

ABSTRACT

Learning from Demonstrations (LfD) refers to using expert demonstrations combined with the reward information given by the environment to jointly guide the learning of policy in Reinforcement Learning. Previous LfD methods usually assume that provided demonstrations are perfect., while in real-world applications, demonstrations are often collected from multiple sources, which may contain imperfect ones. In this work, we aim to deal with the latter situation, i.e., *Learning from Imperfect Demonstrations (LfID)*, where demonstrations only include trajectories with state-action pairs. To this end, two challenges need to be solved: evaluation for the demonstrations and calibration for the bonus model. Both challenges can be more severe in sparse reward environments, since the exploration problem will appear while learning. In this work, we focus on bridging the exploration and LfID problems in view of anomaly detection, and further proposing *AGPO* method to deal with these problems. Compared with state-of-the-art methods, empirical studies on some challenging continuous control benchmarks show the superiority of AGPO in this scenario.

KEYWORDS

Reinforcement Learning; Learning from Demonstrations; Imperfect Expert Demonstration

ACM Reference Format:
Zi-Xuan Chen, Xin-Qiang Cai, Yuan Jiang*, Zhi-Hua Zhou. 2022. Anomaly Guided Policy Learning from Imperfect Demonstrations. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 9 pages.

1 INTRODUCTION

Reinforcement Learning (RL) [38] has been widely used in many challenging sequential decision-making tasks, such as autonomous vehicle [11, 21, 34], video game playing [3, 8, 31, 46] and robotic control [14, 22], and has achieved significant success. They mainly focus on learning a desired policy through interactions with the environment and the feedback of reward signals. Besides, if there exist some expert demonstrations, we can use them to speed up the policy learning. This kind of methodology is so-called Learning from Demonstrations (LfD) [1, 35]. LfD methods use additional demonstrations collected from experts to initialize the policy, establish additional constraints or add regular terms to the objective to guide the training of the policy, so as to reach the potential high reward area faster. Besides, LfD is different from imitation learning [16, 49] and offline reinforcement learning [18, 23], in which during the training process, the agent can interact with the environment and obtain the true reward at the same time.

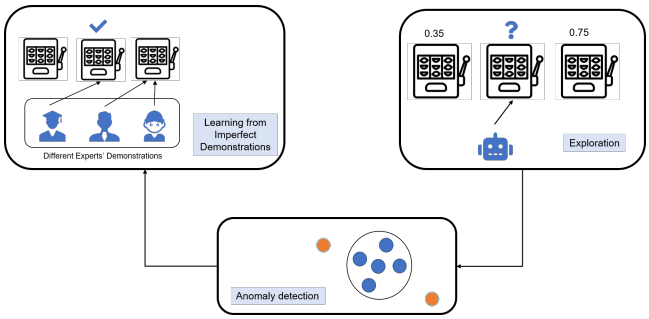


Figure 1: Overview of the methodology of our approach. Most of the off-the-shelf exploration methods can be generalized to the anomaly detection process, and this process can be used to solve the learning from imperfect demonstrations problem.

Previous LfD methods usually assume that the expert demonstrations are collected from an optimal policy, which is very costly in real-world applications due to the demand for high-quality expert demonstrations. So it is necessary to consider the situation where the demonstrations are collected from different experts with distinct qualities. This learning problem is so-called *Learning from Imperfect Demonstrations (LfID)*. Under this setting, traditional LfD methods may be affected by poor quality demonstrations, which may mislead the policy into local optima, and further obtain a worse policy than standard RL methods without additional demonstrations.

In this work, we seek to solve the LfID problem without using any additional information. One of the most challenging problems under LfID is that it is difficult to distinguish demonstrations quality without any other supervision. Therefore, previous work assigns the same importance to all demonstrations can hurt the original policy learning. In the imperfect demonstration learning scenario, high-quality demonstrations can provide positive guidance throughout the training process, while low-quality demonstrations may mislead the policy into sub-optima. Therefore, we need to distinguish the quality of the demonstrations. Besides, when exploring the environment, there exist many collected samples that previous

*Corresponding author. The first two authors contributed equally. Xin-Qiang Cai was a master student at Nanjing University, and now is a phd student at the University of Tokyo.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

methods did not make full use of. So we aim to use these samples to fix the misleading issues of the imperfect demonstrations.

More specifically, when dealing with exploration problems, a widely used assumption is that "novel" states represent "good" states [5, 40], which motivates the policy to explore more unvisited states with potentially higher rewards. Most of these methods guide the policy to explore high-reward areas by estimating the anomaly degree of current states compared with previous experience, which means that this process can actually be generalized to the anomaly detection process, i.e., if a roll-out sample is more anomalous, the more likely it would be a "novel" sample. Moreover, some anomaly detection methods can be utilized as the one-class classification to estimate the probability of the current sample belonging to the expert demonstrations. In such a case, we can use homogeneous models with the same metrics to distinguish the demonstrations and the roll-out samples meanwhile calibrate the model from demonstrations, in order to guide the policy learning better. Inspired by this idea, we propose a novel method named Anomaly Guided Policy Optimization (AGPO) to deal with LfID problems, with anomaly detection metrics. The overview of this idea is illustrated in Figure 1.

We evaluate AGPO on multiple challenging continuous control tasks including MuJoCo [43] and Deepmind Control Suite [42]. Extensive experimental results demonstrate the stability and effectiveness of AGPO, which can surpass existing LfD methods, and achieve high performance even when the proportion of high-quality demonstrations is very small.

The contributions of our work can be summarized as follows:

- We propose an anomaly-guided methods to solve the LfID problem named AGPO. Our method can guide the policy to obtain promising results with imperfect demonstration.
- We combine expert demonstrations with samples collected online to continuously modify the bonus function during the training process. As far as we know, we are the first to adopt this paradigm to design bonus functions.
- Experiments on multiple continuous control tasks show that our method can achieve better performance than the previous methods when the source and quality of expert examples are diverse.

2 RELATED WORK

Learning from Demonstrations (LfD) is a learning variety of reinforcement learning by adding some prior demonstrations to speed up the policy learning. Previous methods usually use the provided demonstrations in two ways. The first type of method uses the given demonstrations to initialize the model, such as behavior cloning [4]. Such methods allow the model to get relatively high rewards at the beginning of training. But when the number of demonstrations is small or the model encounters a state that differs greatly from the distribution of demonstrations, such methods may fail to guide the model to make good decisions. DQfD [15] and DDPGfD [45] proposed in recent years also belong to this type of method. Different from previous behavior cloning methods, these two methods adopt off-policy sampling, and place the demonstrations in the buffer as the augmented the data for the subsequent policy update process. So they also need the reward signals in demonstrations.

Another type of methods use the "bonus" as an additional reward incentive to guide the policy learning, by measuring the similarity between the samples from the learner and the expert. Most of the recent works on LfD have adopted this paradigm to fully utilize the demonstrations. Kang et al.[20] proposed POfD which utilizes Generative Adversarial Networks (GAN) [13] to measure the discrepancy. They train a discriminator to distinguish between the expert demonstrations and the samples collected from the environment, and train a policy to fool the discriminator as much as possible. Brys et al.[6] and Wu et al.[48] use demonstrations to establish a potential function and adopt the reward shaping [28] method for value iteration. We refer to this type of method as the "bonus-based" method. Our method also belongs to this category.

While in reality, the demonstrations may be collected from multiple experts with different levels, namely imperfect demonstrations. We refer to the LfD problem using this type of demonstration as Learning from Imperfect Demonstrations (LfID). The setting of LfID is more realistic yet challenging, for simply matching the sample distribution may mislead the agent into imitating low-quality demonstration and falling into local optima. In recent years, there have been some studies on LfID. Jing et al. [19] proposed to use imperfect demonstrations as constraints. Rather than forcing the agent to completely imitate the demonstrations, their method only uses the true reward to update the policy when the distribution of samples from the current policy is relatively similar to that from the demonstrations. Gao et al. [12] proposed Normalized Actor-Critic (NAC), which is an off-policy actor-critic method using demonstrations as initialization, and it also needs the true rewards in demonstrations, as in DQfD and DDPGfD. Besides, some other studies use imperfect demonstrations in the imitation learning scenario [41, 48], but such methods usually require additional information about the quality of the demonstrations or assumptions about the composition of the samples. Under the setting of LfID, we do not make assumptions about the composition of the demonstration, and no additional information other than the standard environment. We committed to tackling this challenging setting in this work.

Another related branch of our work is anomaly detection [10, 24, 25] which refers to the problem of finding patterns in data that do not conform to expected behavior. Also it has achieved great success in many domains, including intrusion detection [17], healthcare [33], and new class detection [9, 27]. Besides, some recent works studied detecting anomalies using RL methods [26, 30]. But here we focus on improving the LfID policy learning with the anomaly detection methods instead. And the following sections will show how.

3 PRELIMINARIES

LfID problem is a variety of the reinforcement learning problem, meanwhile, the reinforcement learning process is usually abstracted into a Markov Decision Process (MDP), which can be represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma, \mu)$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the set of all possible states and actions respectively, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function measuring the quality of a transition (s_t, a_t, s_{t+1}), $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the transition probability conditioned on state-action pair, $\mu : \mathcal{S} \to [0, 1]$ is the initial state distribution and $\gamma \in [0, 1]$ is the discount factor.

The goal of reinforcement learning is to find a policy that maximize the expected cumulative reward shown in Equation (1):

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right], \tag{1}$$

where Π is the set of all possible policies. The setting of LfD differs from RL in that additional expert demonstrations are given to guide the learning process. We denote each set of the demonstrations as a set of state-action pairs $\mathcal{D} = \{(s_1, a_1), (s_2, a_2), ..., (s_n, a_n)\}$. The "bonus-based" LfD methods use demonstrations to set up bonus function to guide the agent, which is denoted as $B(s, a, s')$ here. The target of the "bonus-based" method is to maximize the expectation of sum of true reward and bonus, which is expressed as follows:

$$\pi_B^* = \underset{\pi \in \Pi}{\operatorname{argmax}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t) + B(s_t, a_t, s_{t+1}))\right]. \tag{2}$$

Another concept is occupancy measure [39], which is defined as Equation (3) to characterize the state-action pairs that an agent encounters when navigating the environment with policy π:

$$\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi). \tag{3}$$

It has been proved that there is a one-to-one matching between policy π and occupancy measure ρ. That is to say for any possible occupancy measure ρ, there is only one unique policy π that generate this occupancy measure.

Furthermore, the imperfect demonstration is an important concept in our work, which refers to the demonstration collected from N different experts of different levels. N is the total number of experts. Non-optimal demonstrations are gathered from experts that correctly understand the task but cannot operate optimally.To better guide the training, we will ensure that part of the demonstrations are from high-quality experts. But we do not make other additional assumptions about the composition of the demonstrations. For each expert policy, we collect a set of state-action pairs as demonstrations, denoted as $\mathcal{D}_i = \{(s_{i1}, a_{i1}), (s_{i2}, a_{i2}), ..., (s_{in_i}, a_{in_i})\}$, where n_i is the number of state-action pairs collected from expert i. The imperfect demonstration used in our work is the union set of all demonstrations, denoted as $\mathcal{D} = \bigcup_{i=1}^{N} \mathcal{D}_i$. We denote the proportion of the samples generated by each expert as p, $0 \le p_i \le 1$, $\sum_{i=1}^{N} p_i = 1$, the occupancy measure corresponding to each expert's strategy is denoted as ρ_i. The occupancy measure of the imperfect demonstration $\rho_{\text{ID}}(s, a)$ is represented in Equation (4).

$$\rho_{\text{ID}}(s, a) = \sum_{i=1}^{N} p_i * \rho_i(s, a) \tag{4}$$

We denote the policy corresponding to ρ_{ID} as π_{ID}. It is obvious that π_{ID} is non-optimal. Thus, under the setting of LfID, the policy will be misled if it naively follows the imperfect demonstrations.

4 PROPOSED METHOD

Previous works tend to learn the policy by optimizing the similarity between current policy and expert policy [19, 20, 47], which is depicted by the occupancy measures. One of the most representative methods of this learning style is Policy Optimization from Demonstrations (POfD) [20], which gives an additional bonus based on the discrepancy of occupancy measure. The objective of POfD is:

$$\begin{aligned}\pi_{\text{POfD}} = \underset{\pi \in \Pi}{\operatorname{argmin}} \max_{D} -\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t (R(s_t, a_t) + B(s_t, a_t, s_{t+1}))\right] - \lambda_1 H(\pi) \\ + \lambda_2 \mathbb{E}_\pi \left[\log(D(s, a))\right] + \mathbb{E}_{\pi_{\text{E}}} \left[\log(1 - D(s, a))\right],\end{aligned} \tag{5}$$

where $H(\pi)$ is the causal entropy regularizer, π_{E} is the expert policy used to generate demonstrations. The discriminator $D : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which tries to distinguish state-action pairs generated by trained policy and expert, is the core of the method. In order to achieve this objective, POfD uses -$\lambda_2 \log D(s, a)$ as a bonus to guide the policy learning. This method can get good results under the setting of LfD, but when it comes to the setting of LfID, this method may suffer from the following two problems:

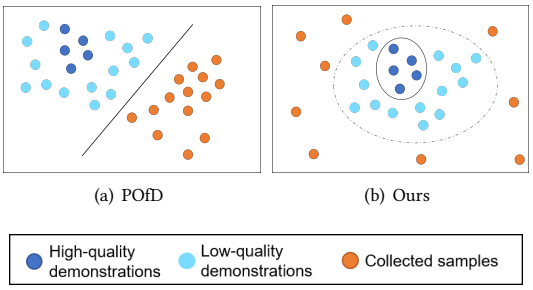


Figure 2: Comparison of our method and POfD. The black line divides the high bonus area from the low bonus area. In Figure (b), dashed line represents the dividing line generated by the sample density of the expert demonstrations, and the solid line represents the dividing line after revise.

- As shown in Figure 2(a), low-quality demonstrations are treated equally as high-quality ones. Under the setting of LfID, giving the same high bonus to both high and low quality samples will be misleading.
- When the given data is not similar to neither the collected sample nor the expert sample, it is still possible to receive a higher bonus.

These problems prevent the previous LfD methods from performing well under the LfID setting.

4.1 Anomaly Based Distribution Estimation

To address the above two issues, we propose a novel bonus function by estimating the anomalies of state-action pairs to guide the learning of policy. The anomalous degree of gathered samples can measure the quality of current data. Meanwhile, it can measure whether the gathered samples belong to expert data (by regarding the demonstrations as normal data and other samples as anomalous data). Based on this idea, we can use the homogeneous anomaly detection model to solve both two tasks simultaneously and make them calibrate each other.

Intuitively, from the definition of occupancy measure in Equation (3), the larger the value of occupancy measure, the more likely

a certain state-action pair is to be accessed by the corresponding policy. Under the case of undiscounted MDP, where the discount factor γ equals 1, the occupancy measure in Equation (3) can be simplified as:

$$\rho(s,a) = \pi(a|s) \sum_{t=0}^{\infty} P(s_t = s|\pi). \tag{6}$$

From the equation, we can see that in undiscounted MDP, the occupancy measure ρ represents the probability of a state-action pair being accessed by its corresponding policy π. Therefore, we can regard the occupancy measure as a distribution in the state-action space, which is usually estimated by sampling multiple times in the environment using the corresponding policy. Obviously, the more considerable the occupancy measure corresponding to a state-action pair, the higher probability this pair would appear with the policy π.

With this measurement, we propose to directly model the distribution of the state-action pairs in expert demonstrations and guide the policy learning by rewarding the pairs that follow such distribution. As shown in Figure 2(b), the region within the black dashed line indicates the area that follows the distribution of demonstrations, and the rest of the areas are out-of-distribution (OOD) region (orange dots). Anomaly detection methods are commonly used to address the OOD detection problem. The data following the distribution region is called inlier, and the data out of distribution region is called anomaly or outlier. The output of the anomaly detector is a real value between 0 and 1, measuring the anomalous degree of a sample. We denote the anomaly detector we use as $A_{\mathrm{d}} : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$, the higher the output value is, the more abnormal the corresponding sample will be. In such a case, the estimated distribution can be denoted as:

$$\widehat{\rho}_{\mathrm{ID}}(s,a) = \frac{1 - A_{\mathrm{d}}(s,a)}{\sum_{(s',a') \in \mathcal{S} \times \mathcal{A}} [1 - A_{\mathrm{d}}(s',a')]}. \tag{7}$$

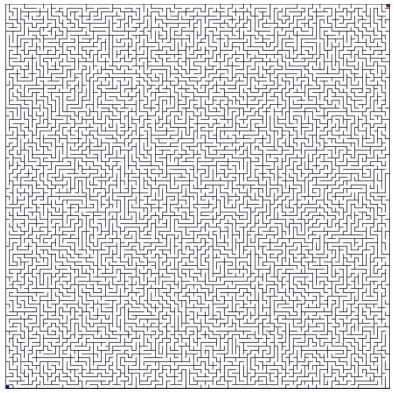
On the other hand, since the denominator is fixed all the time, the bonus can be directly assigned as the numerator:

$$b_{\mathrm{d}}(s,a) = 1 - A_{\mathrm{d}}(s,a). \tag{8}$$

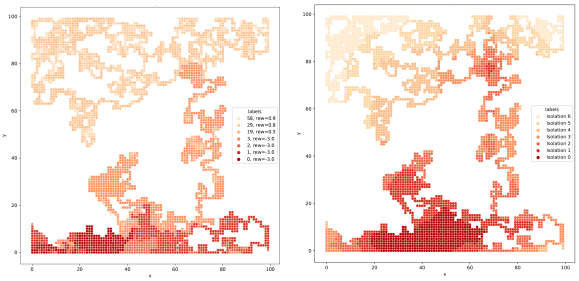
4.2 Anomaly Re-weighted Bonus

As discussed above, the key issue under the LfID setting is how to guide the agent to pay more attention to high-quality demonstrations. We found that the samples collected during the training process were not fully utilized. Therefore, we propose to calibrate the distribution estimation in the Equation (8) using the collected samples to better depict the feature of high-quality demonstration.

To motivate the agent to explore more states to reach the potential high reward area, a commonly used assumption is that a "novel" state tends to be a "good" state. "Novel" states refer to states that are rarely accessed [5, 40] or exhibit different characteristics from the previous states [7, 32]. In short, states that are out-of-distribution from previously visited states are "novel" states as we need. Our method uses this assumption to re-weight the bonus function in Equation (8). Since our bonus function is designed according to the distribution of demonstrations, this weight can be regarded as a modification of sample distribution to better describe the feature of the high-quality policy.



(a)



(b) (c)

Figure 3: (a): the map of the maze. The lower left corner is the entrance and the upper right corner is the target. (b): the reward map of the trajectories. The lighter the color is, the higher quality of the trajectory will be. (c) the score map of the trajectories. The lighter the color is, the more anomaly of the trajectory will be.

On the other hand, the anomalous degree can properly depict how "novel" the current state is. So our method uses previously visited states to build an anomaly detection model and use it to evaluate the anomalous degree of the current state. The higher the output value of the model is, the more anomaly the corresponding state would be. This property is exactly what we desire to evaluate the quality of samples. So we use this result to modify the bonus function in the Equation (8), to give a higher bonus to the "novel" states and reduce the bonus of the "old" states.

To show how the anomaly detection model can evaluate the quality of states, we collect several trajectories with different quality in a 100×100 Maze environment (Figure 3(a)) and build an Isolation Forest [25] anomaly detector on these samples and use it to score them. As shown in Figure 3, Figure 3(b) shows the collected trajectories, lighter color means higher quality of trajectory, while Figure 3(c) shows the score given by anomaly detector, lighter color means higher score. The quality of trajectories matches perfectly with the anomaly score, which demonstrates that the anomaly detection method can ideally reflect the difference of sample's quality.

Here we use the "quality weight" as the measure to re-weight the bonus model, denoted as w_{q}. Meanwhile, $A_q(s) : \mathcal{S} \rightarrow [0,1]$ denotes a state based anomaly detector. For a transition (s,a,s'), the quality weight can be estimated by its anomalous degree of the

next state s':

$$w_{\mathrm{q}}(s, a, s') = A_{\mathrm{q}}(s'). \tag{9}$$

Combining Equation (9) and Equation (8), the total reward for our method is shown in Equation (10). $B(s, a, s')$ is the bonus function, while $T(s, a, s')$ is the combined reward for a transition. β is a hyper-parameter balancing the reward given by environment and bonus:

$$T(s, a, s') = R(s, a) + \beta B(s, a, s'), \tag{10}$$

where

$$B(s, a, s') = w_{\mathrm{q}}(s, a, s') \times b_{\mathrm{d}}(s, a) = A_{\mathrm{q}}(s') \times [1 - A_{\mathrm{d}}(s, a)]. \tag{11}$$

Then we can use policy optimization methods such as Proximal Policy Optimization (PPO) [37] to update the policy using $T(s, a, s')$ as reward. We name our method Anomaly Guided Policy Optimization(AGPO) as we guide the policy learning and the bonus calibration with the anomaly detection methodology. The training procedure of AGPO is illustrated in Algorithm 1.

Algorithm 1 Anomaly Guided Policy Optimization(AGPO)

Given demonstrations $\mathcal{D}$, Environment *env*
Initialize Policy π
Update frequency for A_{q}: t_{update}
A_{d} =Build_Detector($\mathcal{D}$) ▷ Build anomaly detector using demonstration
$\mathcal{D}_q$=Sampling(env, π)
A_{q} =Build_Detecor($\mathcal{D}_q$) ▷ Initialize quality weight using samples collected by initialized policy π
$\mathcal{D}_q = \emptyset$
for $iter = 1; iter <=$ num_update; $iter++$ **do**
  s_0 =Initialize_Env(env)
  for $t = 0; t <=$ num_samples; $t++$ **do**
    $a_t = \pi(s_t)$
    $s_{t+1}, R_t = env.step(s_t, a_t)$
    Calculate $T(s_t, a_t, s_{t+1})$ by Equation (10)
    Update policy π using Policy Optimization method (e.g., PPO) with $T(s_t, a_t, s_{t+1})$
    $\mathcal{D}_q = \mathcal{D}_q \cup \{s_{t+1}\}$
  end for
  if $iter\%t_{\text{update}} == 0$ **then**
    A_q =Update_Detector($\mathcal{D}_q$)
  end if
end for

4.3 Discussion

Our method is closely related to the problems of LfD and exploration in RL. Here we discuss the connection between our method and these two learning problems.

Learning from Demonstration. Occupancy measure matching is a useful method to fully utilize the demonstrations under the setting of LfD. Policy Optimization from Demonstrations(POfD) [20] provides bonus to the policy based on how well the policy matches the demonstrations. Their method requires expert strategies to be optimal. But when this requirement is not met, the bonus will often mislead the trained policy. To solve the problem of imperfect demonstration, Jing et al.[19] only uses samples as constraints: When the occupancy measure of the current policy is relatively similar to that of the demonstration, only real rewards will be used to update the policy. This method requires the careful design of hyper-parameters to set up constraints. Instead, in this work we take advantage of the connection between the demonstrations and the collected data. Our method circumvent the poison of imperfect demonstrations more naturally, meanwhile calibrating the bonus function throughout the whole learning process to keep it abreast of the continuously updated policy.

Exploration in RL. The assumption that "novel" states tend to be "good" states, as the re-weight term in our bonus function, is commonly used in exploration RL methods [7, 32]. In the environments with discrete states, count-based methods such as Upper Confidence Bound (UCB) [2] are commonly used to give a higher bonus to states that are rarely visited. When the state space is large or continuous, previous methods either discretize the state [29, 40] or measure the uncertainty of their estimated models to the current state [7, 32]. These methods encourage the agent to visit more anomalous (compared with the collected ones) states, which can be modeled by off-the-shelf anomaly detection methods, to reach rarely visited states in discrete space or uncertainty states in continuous space. This is closely related to our anomaly re-weighting term. Although the exploration methods can guide the agent to reach the potential high-reward areas, the exploration direction is often random and with high variance. On the other hand, with the help of provided demonstrations, the agent can explore the environment more effectively.

Our method combines the characteristics of LfD and exploration, and solves them simultaneously using the classic unsupervised anomaly detection methods, thus solving the LfID problem simply yet effectively. As far as we know, our work is the first to solve the LfID problems by connecting these two research areas.

Table 1: Average return of collected trajectories.

Environment	Source 1	Source 2	Source 3	Source 4
HalfCheetah	5472	2018	1513	1040
Hopper	2579	1251	1115	859
Reacher	-1.8	-3.4	-5.1	-9.0
Swimmer	128.3	116.5	84.8	48.7
Walker2d	6185	3921	1037	455
Cartpole.Swingup_Sparse	825	309	121	10.6
Finger.Spin	954	838	590	293

5 EXPERIMENT

For the LfID problem, we raised three questions. The following experiments will verify whether our method can solve these issues.

- **Q1.** Can our method successfully estimate the quality of demonstrations without additional information?
- **Q2.** Can our methods outperform prior methods given imperfect demonstrations?
- **Q3.** Whether our method is robust with different ratio of imperfect demonstrations?

In order to investigate the above issues, we conducted a number of experiments in seven challenging continuous-control environments, separately *HalfCheetah, Hopper, Reacher, Swimmer*, and *Walker2d* in MuJoCo [43] as well as *Cartpole.Swingup_Sparse* and *Finger.Spin* in

Deepmind Control Suite[42]. We also modified the original MuJoCo environment and designed sparse reward tasks for *Walker2d* and *HalfCheetah* environments to further validate our method.

5.1 Settings

To evaluate the effectiveness of our method, we collected several trajectories from diverse demonstrators of different levels. We obtain the high-quality demonstrator (Source 1 in Table 1) by running PPO [37] on dense-reward environments for 1e8 steps. Meanwhile, to generate low-quality demonstrations, we use agents that are insufficiently trained (less than 1e7 steps with PPO) as part of the imperfect demonstrators. To make the demonstrators more diverse, we also run TRPO and DDPG for 1e7 and 5e6 steps to generate imperfect policies. After that, we choose three of the above policies that differ greatly from each other in terms of cumulative rewards as low-quality demonstrators (Source 2, Source 3 and Source 4 in Table 1). The average return of trajectories is shown in Table 1.

In order to verify our ideas, we use two anomaly detection models, i.e., One-class SVM [36] and Isolation Forest [25], to build the bonus function. We compare our method with POfD [20], which is the SOTA method under the setting of LfD, and exploration method Random Network Distillation (RND) [7]. We use PPO [37] to update our policy. To better show the impact of different bonus functions on the training process, we also set the naive PPO method with and without behavior cloning pre-training as contenders. For behavior cloning, we initialize the policy for 40 epochs using Adam optimizer with L2-loss and a fixed learning rate of 3e-4. We run 10M steps on *Reacher* and 20M steps on other environments. To reduce the influence of random seeds on the training results, we conduct experiments on 5 different random seeds for each method.

5.2 Modification of Bonus

In order to answer **Q1**, we visualize the bonus given by the policy trained with AGPO. Specifically, we conduct our experiment on *HalfCheetah*, which is an exploration-required environment. We first use imperfect demonstrations to train A_{d}. The composition of the demonstration is shown in Table 1. We collect 5 trajectories from each source and generate an expert demonstration with 20 trajectories. Only 1/4 of the demonstrations are from high-quality expert (Source 1). Then we collect 20 trajectories using various policies to build A_{q}. Finally, we collect another 20 trajectories with different qualities, and calculate bonus on these trajectories.

The results are shown in Figure 4. Figure 4(a) is the t-SNE visualization [44] of the evaluation samples, in which high-quality samples are colored in dark blue. Figure 4(b) is the score given by A_{d} and Figure 4(c) shows the revised bonus given by our method, in which deep color represents high score. We can observe that, since low-quality samples account for a larger proportion of expert demonstration (in Figure 4(b)), low-quality trajectories often get higher scores only from A_{d}. After being re-weighted by "quality weight" A_{q}, high-quality trajectories can obtain higher bonus than low-quality ones. This result depicts that A_{d} indeed has ability to cover the original occupancy measure distribution of demonstrations, while the quality weight A_{q} is able to differ the quality of A_{d}, in order to calibrate the bonus model and thereby effectively leading the policy imitating the high-quality parts.

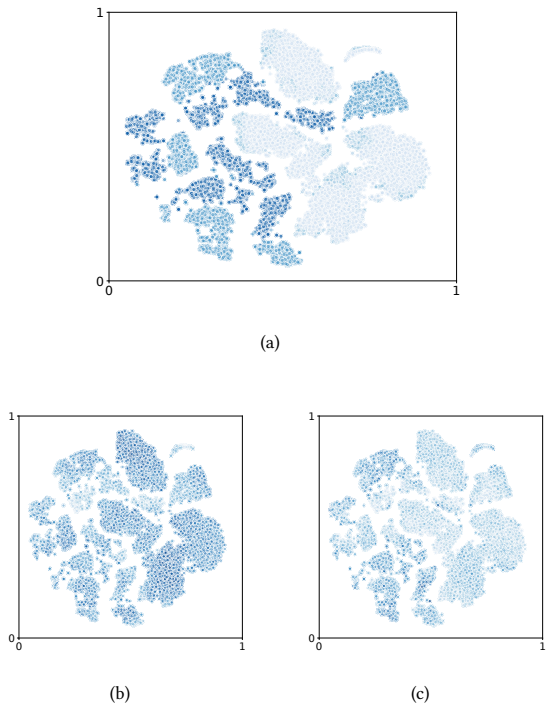


Figure 4: Figure (a) is the t-SNE of samples to evaluate. Figure (b) and (c) are respectively the score based on the sample density and the bonus given by our method. Dark color represents higher score.

5.3 Performance

To answer **Q2**, we design a challenging setting, where only a quarter of the demonstrations are from the high-quality expert. The number of trajectories and samples used in this section is shown in Table 2. In order to further increase the challenge of the task, we refer to POfD [20] and design the sparse-reward tasks for *HalfCheetah* and *Walker2d* environments. We use the same expert trajectories in dense-reward environments to guide the sparse-reward tasks, and use the same cumulative reward as the original task to measure the performance of the policy. For detailed information about the environments and corresponding tasks, please refer to the Section 1 of the Supplementary Material.

The experimental results of all contenders and AGPO with two different anomaly detection methods are shown in Figure 5. From the results, we can observe that the imperfect demonstration setting can greatly harm the performance of POfD. In most environments, POfD performs even worse than the standard RL method PPO. This observation demonstrates that simply matching occupancy measure under the setting of the imperfect demonstration will indeed mislead the policy into following poor-quality experts. The exploration method RND can outperform PPO in some exploration-required environments. But since the direction of exploration is relatively random, RND converges slowly in most of the environments.

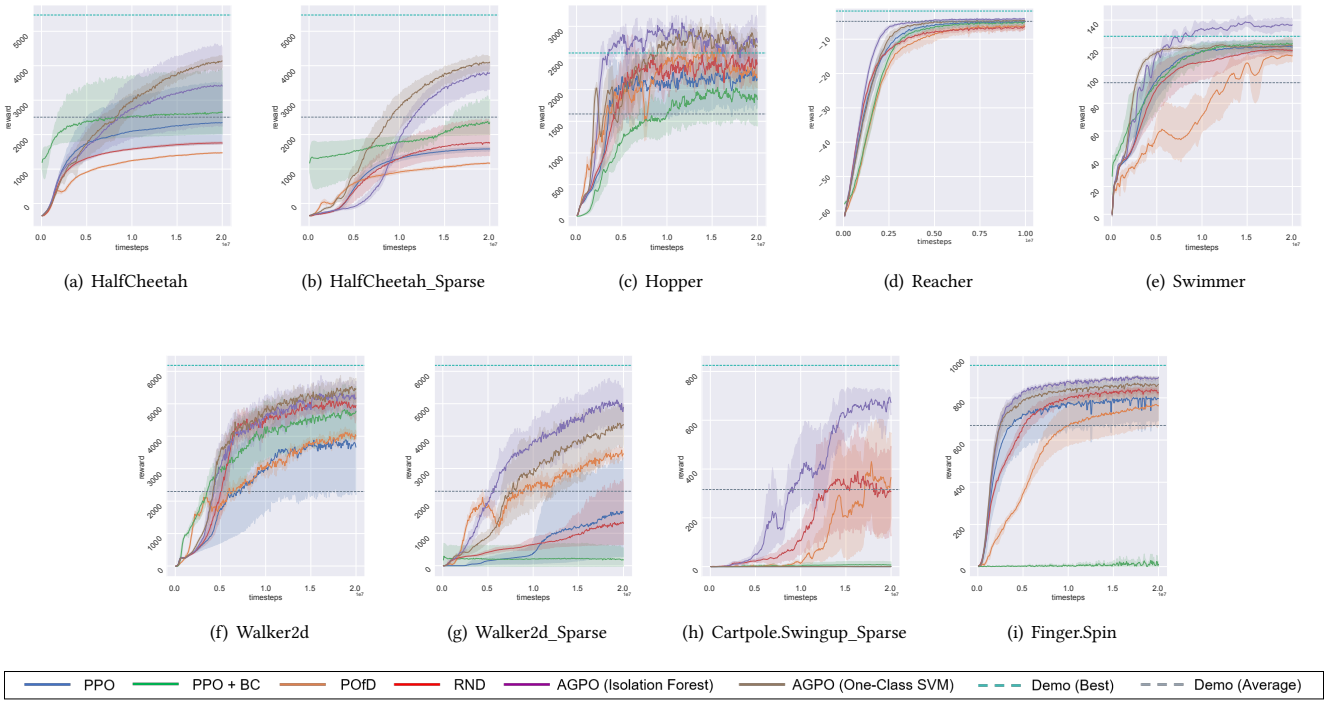


Figure 5: Learning curves of all methods. Shaded regions indicate the standard deviation of 5 seeds.

Table 2: Information about expert demonstrations used for training

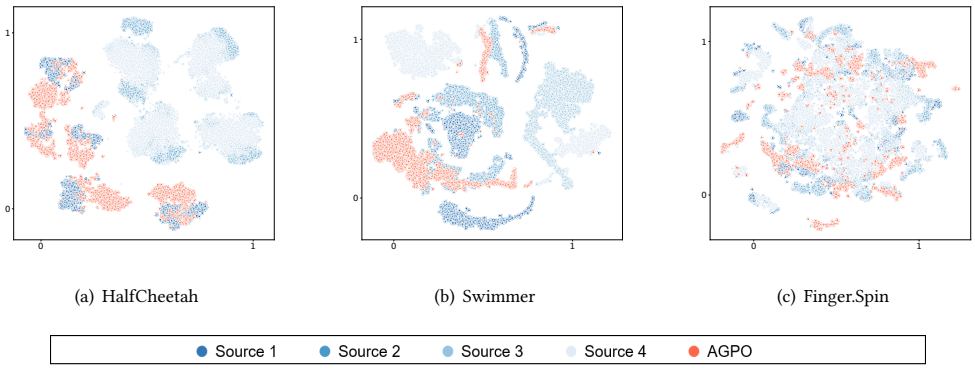
Environment	Source 1	Source 2	Source 3	Source 4	Avg Return
HalfCheetah	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	2506
Hopper	4 trajs, 3900 samples	6 trajs, 3589 samples	8 trajs, 3509 samples	10 trajs, 2144 samples	1616
Reacher	5 trajs, 250 samples	5 trajs, 250 samples	5 trajs, 250 samples	5 trajs, 250 samples	-4.8
Swimmer	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	94.8
Walker2d	5 trajs, 5000 samples	7 trajs, 4606 samples	6 trajs, 5980 samples	10 trajs, 1401 samples	2297
Cartpole.Swingup	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	316
Finger.Spin	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	5 trajs, 5000 samples	669

Our method AGPO surpasses its contenders in terms of convergence speed and performance in most environments. AGPO has reached a level close to high-quality demonstrations in *HalfCheetah, Cartpole Swingup_Sparse, Finger*, and *Walker2d* tasks. In *Hopper* and *Swimmer* tasks, our method even exceeds the best demonstrations. It is worth noting that in *Cartpole Swingup_Sparse, HalfCheetah_Sparse*, and *Walker2d_Sparse* task, the reward is very sparse, the agent can only get a certain reward when it reaches a specific target, which means these tasks are much more challenging than the others. Experimental results show that our method can still guide the policy towards high rewards. This verifies that our method can efficiently utilize imperfect demonstrations even under these challenging environments.

To further reveal how our method uses expert demonstration to guide training, we use the policy trained by our method to collect samples and combine them with expert demonstrations to draw the t-SNE graph. We collected samples on *HalfCheetah*, *Swimmer*, and *Finger.Spin* tasks and randomly select 4000 state-action pairs from each source to generate the graph. As shown in Figure 6, the blue samples are expert demonstrations; deeper color represents higher quality; orange samples are collected by our method. From the t-SNE visualization, the samples generated by the policy of AGPO (orange dots) almost overlap with high-quality expert demonstration (dark blue dots). This result demonstrates that our method can guide the agent to adopt a similar policy to high-quality experts.

5.4 Robustness

To answer **Q3**, we compare our method with other LfD methods using different demonstration compositions. We conduct experiments on *HalfCheetah* and *Cartpole.Swingup_Sparse* tasks. Demonstrations



(a) HalfCheetah (b) Swimmer (c) Finger.Spin

Figure 6: t-SNE graph of expert demonstrations (blue) and samples generated by our method (orange) on three tasks. Samples generated by our methods demonstrate high similarity with high-quality samples.

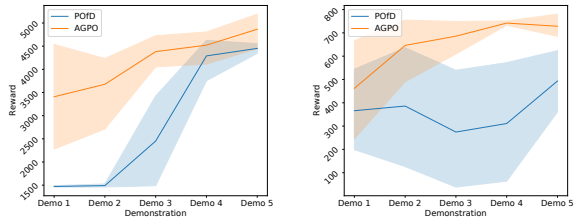
Table 3: Average return for demonstrations with different compositions.

	HalfCheetah	Cartpole
Demo 1	2506	316
Demo 2	2933	341
Demo 3	3525	454
Demo 4	3994	513
Demo 5	5472	825

of different qualities are generated using policies in Table 1. We combined these samples in different proportions and generated five different expert demonstrations for each environment. The average returns for demonstrations of different quality are shown in Table 3, in which the number of state-action pairs in these demonstrations remains the same. The results are shown in Figure 7. The x-axis represents the sample quality, and the more to the right, the higher the sample quality. The y-axis represents the average reward of the trajectories collected by the trained policy. From the results, we can see that the performance of our method surpasses that of POfD on demonstrations of different quality. Meanwhile, with the improvement of demonstration quality, the results of our method become more stable. This shows the robustness of our method, which can achieve high performance regardless of the proportion of imperfect demonstration.

6 CONCLUSION

In this paper, we tackle the challenging learning problem, Learning from Imperfect Demonstrations (LfID), under which current state-of-the-art methods fail. The challenge of this problem comes from the hidden imperfect demonstrations, which cannot be found out directly without other supervision. So we need to properly evaluate the quality of provided demonstrations and calibrate the bonus model from them. Inspired by the anomaly detection methodology in the exploration of RL, we propose a novel approach named



(a) HalfCheetah (b) Cartpole Swingup_Sparse

Figure 7: Experimental results of our method (orange line) and POfD (blue line) on expert samples of different qualities. The abscissa represents expert samples of different qualities, and the ordinate represents the average reward of the trajectories obtained by the policy after the training.

Anomaly Guided Policy Optimization (AGPO) to solve the challenging LfID problem, which can ideally meet the two requirements above by unifying the homogeneous anomaly detection models from demonstrations and roll-out samples. Experimental results under several continuous control environments verify the effectiveness and robustness of AGPO, which outperforms both the average performance of demonstrations and state-of-the-art contenders, with significant gaps. In the future, we expect AGPO can also provide inspiration in other RL problems, such as the exploration-exploitation dilemma of the reinforcement learning community.

ACKNOWLEDGMENTS

This research was supported by NSFC (62176116, 61921006) and the Collaborative Innovation Center of Novel Software Technology and Industrialization. The authors would like to thank Yao-Xiang Ding and the anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[2] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[3] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. 2020. Agent57: Outperforming the atari human benchmark. In *International Conference on Machine Learning*. 507–517.

[4] Michael Bain and Claude Sammut. 1995. A Framework for Behavioural Cloning.. In *Machine Intelligence 15*. 103–129.

[5] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems* 29 (2016), 1471–1479.

[6] Tim Brys, Anna Harutyunyan, Halit Bener Suay, Sonia Chernova, Matthew E Taylor, and Ann Nowé. 2015. Reinforcement learning from demonstration through shaping. In *Twenty-fourth international joint conference on artificial intelligence*.

[7] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894* (2018).

[8] Xin-Qiang Cai, Yao-Xiang Ding, Yuan Jiang, and Zhi-Hua Zhou. 2021. Imitation Learning from Pixel-Level Demonstrations by HashReward. In *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé (Eds.). ACM, 279–287.

[9] Xin-Qiang Cai, Peng Zhao, Kai-Ming Ting, Xin Mu, and Yuan Jiang. 2019. Nearest Neighbor Ensembles: An Effective Method for Difficult Problems in Streaming Classification with Emerging New Classes. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, Jianyong Wang, Kyuseok Shim, and Xindong Wu (Eds.). IEEE, 970–975.

[10] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009), 15:1–15:58.

[11] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*. 2765–2771.

[12] Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. 2018. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313* (2018).

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[14] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*. 3389–3396.

[15] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. 2018. Deep q-learning from demonstrations. In *Thirty-second AAAI conference on artificial intelligence*.

[16] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.

[17] J. Jabez and B. Muthukumar. 2015. Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. *Procedia Computer Science* 48 (2015), 338–346.

[18] Shengyi Jiang, Jingcheng Pang, and Yang Yu. 2020. Offline imitation learning with a misspecified simulator. *Advances in neural information processing systems* 33 (2020).

[19] Mingxuan Jing, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Chao Yang, Bin Fang, and Huaping Liu. 2020. Reinforcement learning from imperfect demonstrations under soft expert guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5109–5116.

[20] Bingyi Kang, Zequn Jie, and Jiashi Feng. 2018. Policy optimization with demonstrations. In *International Conference on Machine Learning*. 2469–2478.

[21] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[22] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

[23] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-Based Anomaly Detection. *ACM Trans. Knowl. Discov. Data* 6, 1 (2012), 3:1–3:39.

[25] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. 413–422.

[26] Xiangyu Ma and Wei Shi. 2021. AESMOTE: Adversarial Reinforcement Learning With SMOTE for Anomaly Detection. *IEEE Trans. Netw. Sci. Eng.* 8, 2 (2021), 943–956.

[27] Xin Mu, Kai Ming Ting, and Zhi-Hua Zhou. 2017. Classification Under Streaming Emerging New Classes: A Solution Using Completely-Random Trees. *IEEE Trans. Knowl. Data Eng.* 29, 8 (2017), 1605–1618.

[28] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, Vol. 99. 278–287.

[29] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. 2017. Count-based exploration with neural density models. In *International conference on machine learning*. 2721–2730.

[30] Guansong Pang, Anton van den Hengel, Chunhua Shen, and Longbing Cao. 2021. Toward Deep Supervised Anomaly Detection: Reinforcement Learning from Partially Labeled Anomaly Data. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 1298–1308.

[31] Zhen-Jia Pang, Ruo-Ze Liu, Zhou-Yu Meng, Yi Zhang, Yang Yu, and Tong Lu. 2019. On reinforcement learning for full-length game of starcraft. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4691–4698.

[32] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. 2778–2787.

[33] Edin Sabic, David Keeley, Bailey Henderson, and Sara Nannemann. 2021. Healthcare and anomaly detection: using machine learning to predict anomalies in heart rate data. *AI Soc.* 36, 1 (2021), 149–158.

[34] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging* 2017, 19 (2017), 70–76.

[35] Stefan Schaal et al. 1997. Learning from demonstration. *Advances in neural information processing systems* (1997), 1040–1046.

[36] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. 1999. Support vector method for novelty detection.. In *NIPS*, Vol. 12. 582–588.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[38] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*.

[39] Umar Syed, Michael Bowling, and Robert E Schapire. 2008. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*. 1032–1039.

[40] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. 2017. # exploration: A study of count-based exploration for deep reinforcement learning. In *31st Conference on Neural Information Processing Systems (NIPS)*, Vol. 30. 1–18.

[41] Voot Tangkaratt, Nontawat Charoenphakdee, and Masashi Sugiyama. 2020. Robust Imitation Learning from Noisy Demonstrations. *arXiv preprint arXiv:2010.10181* (2020).

[42] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690* (2018).

[43] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5026–5033.

[44] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[45] Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817* (2017).

[46] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[47] Yuchen Wu, Melissa Mozifian, and Florian Shkurti. 2020. Shaping Rewards for Reinforcement Learning with Imperfect Demonstrations using Generative Models. *arXiv preprint arXiv:2011.01298* (2020).

[48] Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. 2019. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*. 6818–6827.

[49] Tian Xu, Ziniu Li, and Yang Yu. 2020. Error Bounds of Imitating Policies and Environments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).